

5.4 社会システム障害予見，救助計画立案技術（産業総合技術研究所）

5.4.1 研究の概要と目標

大規模災害・テロ・重大事故などの災害が発生した際の救助活動を円滑に実行するために，災害発生時の状況からの時系列推移をマルチエージェントシミュレーションで推定しながら救助計画を策定する救助計画立案システムの研究開発を行う．時間軸を有する地理情報システムを核とし，そこから得られるリアルタイムでの現場の状況を把握し，その情報を元にマルチエージェントシミュレーションを実行することにより，どのような救助計画が有効であるのかを検証・提案するシステムを構築する．これらのシステムは様々なサブモジュールを統合したシステムとして動作するものとなる．すなわち 1) 交通シミュレータ，人流シミュレータ，構造物シミュレータなど個別のマルチエージェントシミュレータ，2) 統合シミュレーションエンジン，3) 救助計画立案等のための付加モジュール，等である．

5.4.2 平成 16 年度成果の概要

シミュレーションによる情報共有の減災効果検証に必要な基礎データを，共有プラットフォームからシミュレーションシステムに迅速に転送するために，データ入力・検索およびシミュレーション制御のための共通プロトコルの設計を行った．災害に関係する情報には多種多様なものが考えられるため，それらを柔軟に扱いつつ，大量のデータを高速に検索・操作するための枠組みを用意する必要がある．また平常時における自治体業務やインフラ事業者との連携等を考慮した結果，XML の各種標準を用いたプロトコルを設計した．さらに既存の各種システムを迅速に連携させる実験を行うために，そのプロトコルにのっとった情報共有データベースのサンプル実装を行い，機能などを検証した．

さらにシミュレーションについては，中越地震など実際の事例を中心に現状で入手可能な情報の収集を行い，広域交通管制・医療資源分配等を対象に減災効果の検証が可能なテーマの洗い出しを行った．特に医療資源分配については，現状システムの検証とシミュレーションモデルの検討を行った．

5.4.3 情報共有のためデータ形式および管理プロトコルの設計

(1) 災害対策マルチエージェントシミュレーションにおける情報共有基盤の重要性

災害時における情報共有では，平常時には別々に用いられているデータや情報形式を総合して集約していく必要がある．特にデータ形式については互換性や情報システムの相互運用性が壁となり，情報共有を事実上困難にしているのが現状である．

災害のマルチエージェントシミュレーションにおいても同様に，多様なデータ形式にどのように対応するかがおおきな問題となる．マルチエージェントシミュレーションは多数の要素現象を同時にシミュレーションすることで複雑な現象を再現・解析するところに特徴があるが，多数の要素現象のデータをいかに準備するかが大きな問題となる．特に災害シミュレーションにおいては対象が膨大であり，統一的なデータ取り扱いの枠組みが必要になってくる．

そこで本テーマでは災害時の情報共有及びマルチエージェントシミュレーションに必要な情報表現の共通基盤を設計する。図 5.4-1 は本研究で想定している、災害時における情報の流れと共通基盤について示した図である。この図において、左側の部分では災害時の被害状況やシミュレーションに用いる初期値データなど多種多様な情報が収集されて情報共有 DB に情報共有共通プロトコルを介して集約される。そのデータを用いてマルチエージェントシミュレーションにより予測・推定等を行い（中央部）、支援物資配送や交通規制・医療体制などにおける実際の救助計画の立案を支援するものである。

このような基盤を設計するにあたっては全体としての頑健性の確保が重要な課題となる。図 5.4-1 の中央部では1つの情報共有 DB にすべての情報が集約されるように表現しているが、このような集中管理方式は災害時などの非常時には頑健性が確保できない。よって、情報表現共通基盤を設計するにあたっては、GRID 技術や P-to-P 技術など情報インフラの障害に対して頑健な枠組みを取り込むことを前提とする必要がある。

また先に述べているように、災害時には非常に多様な種類の情報を扱う必要がある。これらの情報は平常時は別の目的で用いられていることが多く、そのフォーマットを完全に管理することは難しい。よって、災害時の基盤としては平常時と非常時をシームレスにつなぐデュアルユースの考え方と、汎用性の高い表現形式を選ぶ必要がある。

以下ではこの共通基盤として採用した GGGD プロトコルの概要を紹介し、災害時の情報共有のための拡張について述べる。

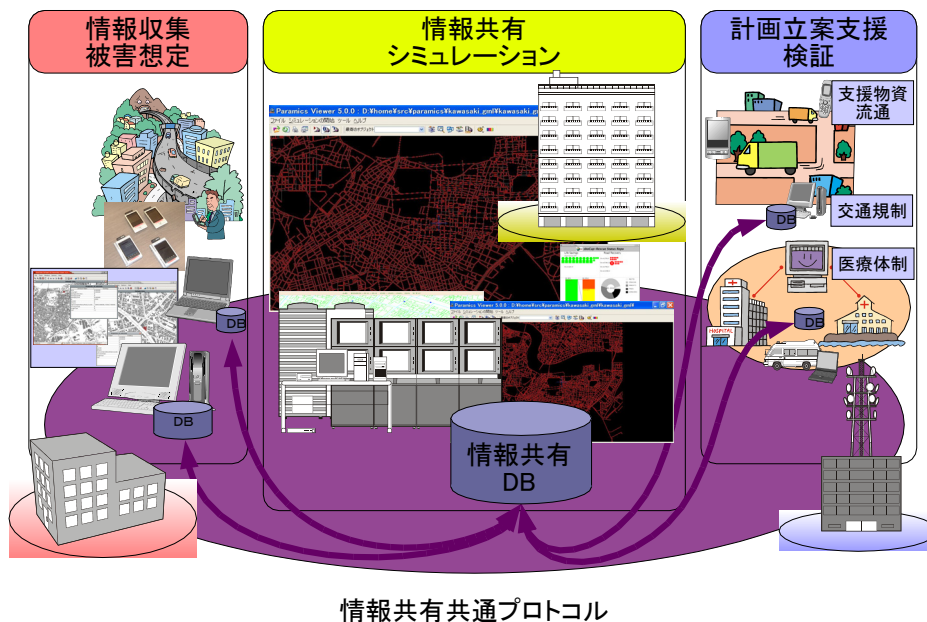


図 5.4-1 災害救助における情報の流れと共通基盤

(2) GGGD (General GeoGraphical Data) プロトコル

前節で述べた、災害時における情報システムの共通基盤に求められる頑健性、汎用性・デュアルユースを満たすものとして、汎用的な地理情報データベースシステム用プロトコルである GGGD

(General GeoGraphical Data) プロトコルを採用し、災害時の情報共有に必要な拡張・改良を試みる。

GGGD (General GeoGraphic Database) プロトコルは、災害情報を含めた汎用的な情報の共有を円滑に行うため、クライアント・サーバ型のサービスを前提としてデータベースアクセスを行うための通信手順である。GGGD プロトコルは TCP/IP による通信を仮定しており、XML をベースとした web サービスとして設計されている。

扱われるデータは各々木構造のデータ構造をもつものとして扱われ、その単位をデータエンタリと呼ぶ。データエンタリには型によって分類され、1つの型に属するデータエンタリはすべて同じ木構造を持つものとする。この木構造は XML Schema により定義され、以下に述べる Register プロトコルによりシステムに登録される。

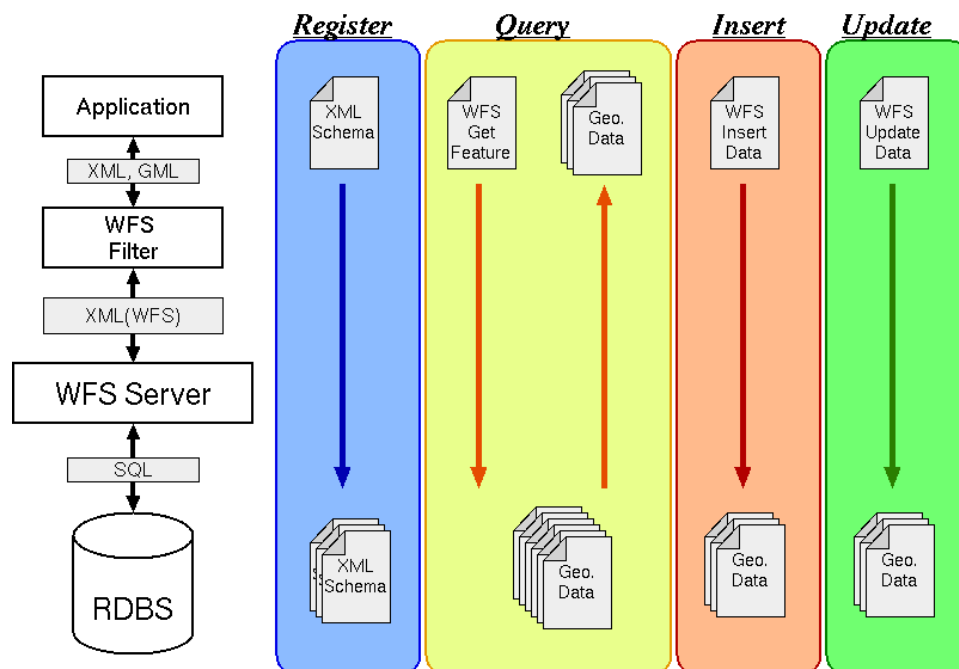


図 5.4-2 GGGD プロトコル

データベースを検索するクエリの形式としては WFS(Web Feature Service)をベースに独自の拡張をしたものを用いる (図 5.4-2 参照)。具体的には、GGGD プロトコルは WFS で定義されているプロトコルのうち、以下のものをサポートする。

- GetFeature:データベースに記録されているデータエンタリの検索を行う。
- Transaction:
 - Insert:データベースに、新たなデータエンタリを追加記録する。
 - Update:データベースに記録されているデータエンタリの内容を変更する。
 - Delete:データベースに記録されているデータエンタリを削除する。
- GetCapabilities:データベースが提供するサービスに関する情報を知らせる。
- DescribeFeatureType:登録されているデータの型に関する情報を、XML Schema の形式で知

らせる。

さらに、データの型の登録のために、以下の拡張プロトコルをサポートする。

- **Register:**データベースに新たなデータの型を定義し、その型のエントリを扱うための設定などを行う。

(3) カプセル化

GGGD で採用している WFS プロトコルは、基本的にはデータベースへのアクセスのみを規定している。しかし先に述べているように、災害時においてはシステム全体としての頑健性が最大の問題となる。また、デュアルユースなどで平常時システムとの連携を考える場合、技術の進歩にあわせてプロトコルが改良された場合、いかに互換性を保つかが問題となる。これらの問題を解決するために、GGGD プロトコルを一段抽象化し、これらの高度な機能を WFS に実装可能にするために、SOAP による WFS のパッケージ化を行った。

具体的には以下のような形式をとる。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns='http://staff.aist.go.jp/i.noda/Rescue/GGGD' >
<SOAP-ENV:Header>
  <protocolVersion>{version of the protocol }</protocolVersion>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  { WFS Protocol Form }
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

形式としては、SOAP のエンベロープにより WFS プロトコルを包んだ形式になっている。SOAP エンベロープのヘッダー部には、独自拡張として<protocolVersion> タグを記述する。これにより、GGGD のプロトコルバージョンを明記し、互換性のチェック及びプロトコルの設定を可能にする。WFS の書式自身は SOAP ボディに格納され処理が行われる。また、通信の途中などでエラーが生じた場合には SOAP の規約に従ってエラーメッセージが返されるものとする。

SOAP を採用したことによるメリットは以下のとおりである。

- GRID および P-to-P 技術を導入する際の整合性が高い。GRID 技術の多くは SOAP による通信をサポートしている。特にヘッダー部に通信の順路や方針を記述することで、単純なサーバ・クライアントシステムでは難しい頑健性を確保することができる。また、WFS による要求を処理すべきサーバを明記する必要がないので、使える計算し現に応じて負荷分散を行うなどスケーラビリティを確保することが可能になる。
- 暗号化などセキュリティの導入が容易である。SOAP のヘッダーのもうひとつの利点はセキュリティ情報を記述することができ、これにより認証や暗号化など柔軟に対応することが可能となる。
- プロトコル管理による互換性維持が容易である。通信自身にバージョンを明記することで、サーバあるいはクライアント側による上位・下位互換の機能を組み込むことが可能である。ただし現在のところ、バージョンの管理方法や分岐バージョンの扱いに付いては未定である。

特に多くの自治体で利用されることを考慮すると、分岐バージョンをどのように整合すべきかを早急に定める必要があり、今後の課題である。

(4) 被覆データの扱い

災害時に必要となるデータには、道路や建造物・人・自動車の状態など点や形状で表される地物と、震度分布や気温・気圧・浸水度など数値やベクトルの分布として表される被覆がある。WFSはこのうち地物データの格納・検索を中心に策定されており、被覆についての取り扱いについては規定されていない。

そこでこの被覆データを取り扱うために、まずプロトコル中で被覆データを表現するために、GML(Geographical Markup Language)の定める被覆データ表現の一部を採用した。具体的には、GML 3.1 の RectifiedGridCoverageType の記述方法に準拠する。ただし、同 Coverage Model の記述方法にはいくつかの冗長性があるため、以下に示す形式のみを扱うものとする。

```
<gml:RectifiedGridCoverage>
  <gml:rectifiedGridDomain>
    <gml:RectifiedGrid dimension="coverage dimension">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>list of lower bounds of each axis</gml:low>
          <gml:high>list of upper bounds of each axis</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>name of the first axis</gml:axisName>
      <gml:axisName>name of the second axis</gml:axisName>
      ...
      <gml:axisName>name of the last axis</gml:axisName>
      <gml:origin>
        <gml:Point>
          <gml:coordinates>position of the origin</gml:coordinates>
        </gml:Point>
      </gml:origin>
      <gml:offsetVector>o@set vector for the first axis</gml:offsetVector>
      <gml:offsetVector>o@set vector for the second axis</gml:offsetVector>
      ...
      <gml:offsetVector>o@set vector for the last axis</gml:offsetVector>
    </gml:RectifiedGrid>
  </gml:rectifiedGridDomain>
  <gml:rangeSet>
    data part
  </gml:rangeSet>
</gml:RectifiedGridCoverage>
```

この形式の中で、*coverage dimension* は、グリッドの次元を表す。例えば、直線上に並んだデータ列の場合は 1 を、2 次元地図上にマップされる場合は 2 を、3 次元空間を埋める場合には 3 を指定する。*list of {lower/upper} bounds of each axis* には、グリッドの各次元の下限・上限を示す。*name of the {first/second/.../last} axis* にはグリッドの次元の名前を記述する。*position of the origin* には原点の座標を指定する。*offset vector for the {first/second/.../last} axis* には、グリッドの各次元の方向ベクトルを記述する。4. よって、グリッドの各点の座標 p は、 $\text{dimension}=3$ 、原点が o 、グリッドの各次元の方向ベクトルが u , v , w とすれば、 $p_{ijk} = o + iu + jv + kw$ となる。

ただし、添字の i, j, k は *list of {lower/upper} bounds of each axis* に指定された範囲を定義域とする。

data part は、`gml:File` もしくは `gml:DataBlock` 形式のいずれかをとるものとする。`gml:File` 形式は以下の通りである。

```
<gml:File>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponent>
        <user-defined tag for the first parameter [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      <gml:valueComponent>
        <user-defined tag for the second parameter [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      ...
      <gml:valueComponent>
        <user-defined tag for the last parameter [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:fileName>URI for the data file</gml:fileName>
  <gml:fileStructure>Record Interleaved</gml:fileStructure>
</gml:File>
```

`gml:DataBlock` 形式は以下の通りである。

```
<gml:DataBlock>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponent>
        <user-defined tag for the first parameter [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      <gml:valueComponent>
        <user-defined tag for the second parameter [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
      ...
      <gml:valueComponent>
        <user-defined tag for the last parameter [uom="urn for unit of the parameter"]/>
      </gml:valueComponent>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:tupleList>list of parameters in CSV style</gml:tupleList>
</gml:DataBlock>
```

以下は、`gml:DataBlock` を用いた、平均気温・気圧のメッシュ上のデータの記述例である。

```
<app:AverageTempPressuer>
  <gml:rectifiedGridDomain>
    <gml:RectifiedGrid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>1 1</gml:low>
          <gml:high>9 6</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>u</gml:axisName>
      <gml:axisName>v</gml:axisName>
      <gml:origin>
        <gml:Point><gml:coordinates>2.0,1.0</gml:coordinates></gml:Point>
      </gml:origin>
    </gml:RectifiedGrid>
  </gml:rectifiedGridDomain>
</app:AverageTempPressuer>
```

```

    <gml:offsetVector>1.0 0.2</gml:offsetVector>
    <gml:offsetVector>-0.2 1.0</gml:offsetVector>
  </gml:RectifiedGrid>
</gml:rectifiedGridDomain>
<gml:rangeSet>
  <gml:DataBlock>
    <gml:rangeParameters>
      <gml:CompositeValue>
        <gml:valueComponent>
          <app:Temperature uom="urn:x-si:v1999:uom:degreesC"/>
        </gml:valueComponent>
        <gml:valueComponent>
          <app:Pressuer uom="urn:x-si:v1999:uom:kPA"/>
        </gml:valueComponent>
      </gml:CompositeValue>
    </gml:rangeParameters>
    <gml:tupleList>2.0, 101.2 5.0, 101.3 7.0, 101.4 11.0, 101.5
                  13.0, 101.6 17.0, 101.7 19.0, 101.7 23.0, 101.8 29.0, 101.9
                  ...
    ...</gml:tupleList>
  </gml:DataBlock>
</gml:rangeSet>
</app:AverageTempPressuer>

```

上記のデータ型は、gml:RectifiedGridCoverageType という名前で ‘common.xsd’ に登録されている。よって、例えば上記の app:AverageTempPressuer というデータ型を扱えるようにするためには、下のような定義を Register プロトコルにより登録すればよい。

```
<element name="AverageTempPressuer" type="gml:RectifiedGridCoverageType"/>
```

(5) 添付形式による大規模データの取り扱い

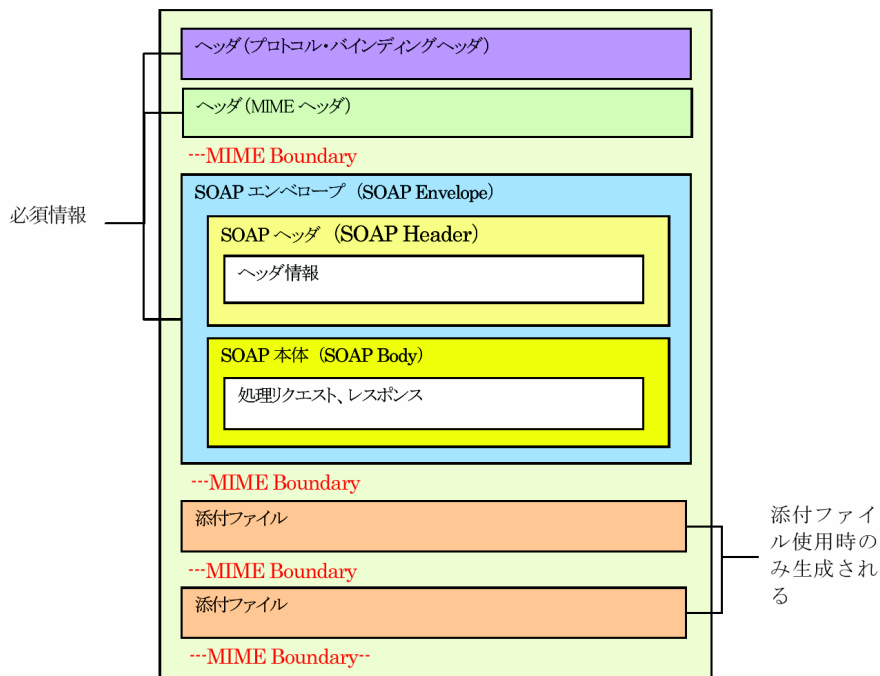


図 5.4-3 MIME による添付形式

上記の被覆におけるファイル形式や地図データにおける全道路情報などはかなり大量のデータ

となりえる。XML ではその冗長性のため、大量のデータを処理する上ではオーバーヘッドが大きくなりすぎることがある。このオーバーヘッドを解消し、比較的能力の低い計算機でも効率よくデータの転送や処理を行わせるために、GGGD に添付形式の表現を導入する。

添付の方法としては、SOAP Message with Attachments に準拠した MIME (Multipurpose Internet Mail Extension) 形式を採用する。具体的には、図 5.4-3 に示すような形式を採用する。

(6) 今後の課題

本年度は災害時の情報共有及びそれを用いたマルチエージェントシミュレーションの観点から災害時の情報共有形式について見当を行い、共通基盤となる標準プロトコルを設計した。しかし、このプロトコルはまだ基本的な部分にとどまっております、具体的な応用プログラムを作成していく上では、以下のような課題が残されている。

- 応用スキーマの策定。本年度設計したプロトコルでは、実際の災害をどのような XML で表現すべきかについては規定しない汎用の枠組みになっている。しかし具体的に相互に連携できるシステムを構築していく上では道路や建物、人的被害についてどのような構造をもって表現すべきかを決めていく必要がある。
- 被覆データの検索プロトコル。現在の GGGD は地物を検索することを主眼として設計されている。しかし実際の被害データやシミュレーションに必要なデータには被覆型のものも多く、そのためのデータベースアクセス方式、特に検索の指定の仕方が重要となる。これについては GML 等において WCS (Web Coverage Service) が策定されてきており、それに準拠したプロトコルを組み込む必要がある。
- 参照実装システムの作成。情報共有を行う複合システムを効率よく作成していく上では、標準プロトコルに準拠したシステムを 1 つ、参照システムとして実装・提供する必要がある。これについてはすでに存在する関係データベースの地理検索機能と簡易言語によるミドルウェアを用いたシステムが稼動しており、これをもとにプロトコルに完全準拠したものを早期に構築することが望ましい。

5.4.4 医療資源配分計画支援システムの検討

(1) 目的

災害時医療は、医療資源の容量を越えて外傷患者が発生するため、通常の救急外傷医療現場以上に、適切な治療によって避け得る外傷死が多く発生する。本研究は、情報共有とマルチエージェント・シミュレーションによって、避け得る外傷死を最小限に抑えて救命率を最大化するような医療資源配分計画を支援するものである。

(2) 従来の災害時の医療情報の扱いについてのサーベイ

a) 既存の情報共有システム

広域災害救急医療情報システム[1]が、すでに各県において運用されている。このシステムは、

被災医療機関が患者転送要請情報を、支援可能医療機関が受け入れ可能患者数などの支援情報を登録し、その情報を国・都道府県担当課が、情報を活用する、というものである。システムが扱う情報は主に患者の収容と搬送に関するものに限られる。自治体対策本部からの救護スタッフ派遣要請情報や、被災地外部への医療チームの要請や受け入れなどの情報は扱っていない。災害時、病院は受け入れ可能ベッド数以上に、中軽傷患者を受け入れるのが実態であり、実質的受け入れ容量は医療資源の人的要素に依存するが、これは情報共有されていない。

b) 指揮・管理系統の並列性

災害発生時において、情報収集・救護活動・避難管理などを行うのは自治体に設置される対策本部である。当然、要救護者を瓦礫の下から救出する消防の情報なども、対策本部が最終的に管理している。一方、医療機関は各地域の医師会によって組織化されている。災害時においても、自治体区域内の医療機関は対策本部の管理下に置かれるわけではなく、医師会が統括している。重症患者の搬送先なども医師会が管理する。これによって具体的に生じる問題は以下の様である。

c) 現場救護所と広域医療の独立管理

災害時、救出活動が行われる被災地において応急医療を行う救護所は、地域病院により自律的集合ないし保健所の要請による広域医療機関・災害拠点病院からの医療スタッフ派遣によって運営される[1]。医療機関からのスタッフ派遣は、自治体と医師会によって平時より締結されている災害時の医療救護活動についての協定に基づいて行われる。組織された救護所の地域間のバランスなどは、現場の状況を被災地に最も近いところで把握している自治体保健所が調整する。一方、救護所から広域医療機関・災害拠点病院への収容先決定は、医師会が管理している。

ここで問題となるのは、現場救護所への人的資源と、救護所に人的資源を貸し出している医療機関での医療行為に従事する人的資源が、救命率最大化のために最適であるかどうかの判断がつきにくい、ということである。その理由は以下である。

- 災害医療において、現場での応急医療と後方での医療は対象とする患者も目的も違う。現場では、重症患者を応急処置して搬送できるようにすることが優先され、後方では救命のための高度な医療が行われる。現状では、被災現場での患者発生数に基づいて救護所への人的資源配置がスケジューリングされている。
- 災害においては時系列的に状況が推移し、それに伴い救護所で必要とされる医療資源と後方広域医療機関で必要とされる資源の適切な配分は変化する。大規模災害では一般市民による自主的搬送も行われるため、本来、救護所で最初のトリアージと応急医療を受けるべきところを、直接広域医療機関に搬送される傷病者も多い。現状では保健所により、救護所間の人的資源配分は行われているが、一旦救護班として派遣された人的資源の配置見直しは行われていない。

現時点で医療機関に導入されている広域災害救急医療情報システムは、各医療機関が提供可能もしくは不足している医療資源のリストと援助要請情報のリストをセンターにアップロードするだけのものであり、医療機関側の相互搬送を支援しているに留まっている。提供可能な資源と需

要のマッチングは、システムによって支援されておらず人間によって行われている。利用可能な医療スタッフの情報や自治体が得た被害情報は共有されておらず、災害時系列に伴う医療資源の再配分に役立つものにはなっていない。また、人的資源配分調整による救命率への影響を評価する手法は現時点で存在しない。

(2) 災害医療シミュレーション・モデル

前節までに述べた、従来の医療情報の扱いの問題点と、救命率評価手法の欠落を埋めるため、本研究では救命率向上のためのシミュレーション・モデルを構築した。

従来、外傷治療のシミュレーション・モデリングにおいては、病院の医療資源の容量を越えるような災害事例を扱っておらず、シミュレーションの目的は、患者の治療までの待ち時間を平均して減少させるために、トリアージ基準の作成や搬送ポリシーの設定を支援するものであった[3]。外傷患者の初見における重症度分類は、救急医学においても分析が困難であり、救命率をターゲットとした医療シミュレーションは行われていない。

a) 外傷患者モデルの構築

避け得る外傷死を最小限にすることを目的とした医学における重症度判定は、外傷の多様性などから困難が伴うが、大規模災害のように個々の災害に特徴的な外傷が多数発生する状況は、遷移確率モデルによって外傷患者の治療プロセスをモデル化することが可能である。

b) 外傷患者の発生

一定時間・一定地域で救助される外傷患者数の確率はポワソン分布で表現される。これを、患者の救助される時間間隔で表せば、以下の指数分布となる。

$$P(x) = \exp(-x/a)/a \quad (a = \text{平均間隔}, x > 0) \quad \dots\dots \text{(式 5.4-1)}$$

例えば、1時間に30人の傷病者が救助される場合、 $a = 2 \text{ m}$ となる。

発生した患者の重症度は標準トリアージからランダムに決める。トリアージとは、限られた人的物的資源の状況下で、最大多数の傷病者に最善の医療を施すため、患者の緊急度と重症度により治療優先度を定めることである。

- 赤R： 緊急治療
- 黄Y： 準緊急治療:2,3時間放置可能
- 緑G： 軽症:放置可能
- 黒B： 死亡もしくは治療の見込みがない

トリアージに基づいた治療・搬送プロトコルは、資源の制限にもよるが、一般的に、現場救護所で緊急治療患者を応急処置により容体を安定させ、準緊急治療患者にした上で後方搬送をする。緊急治療患者と治療の見込みのない患者の区別は困難であり、治療プロトコルは明確に決まっていない。阪神淡路大震災では、準緊急治療患者の治療でさえも医療資源を圧迫したため、医療機関においては緊急治療患者の多くは見込みがないと判断された。本研究のモデルでは、現場救護所では緊急治療患者を優先して治療し、後方では準緊急治療患者を優先して治療することとする。

c) トリアージに基づいた確率遷移

患者に対する処置によって、確率的に患者の状態が遷移する。この遷移確率として、病人が発作発症後に死亡に至る時間の確率を表現していることが知られている指数分布を用いる。

患者に対する行為は以下の4種類とする。

- FM:現場救護所における医療行為
- IM:後方医療機関における医療行為
- T:搬送
- L:放置

各行為により、患者の状態遷移にかかる時間の平均時間パラメータは以下のように定める。

- b, c : 治療時の平均死亡所要時間, $g < b < c$

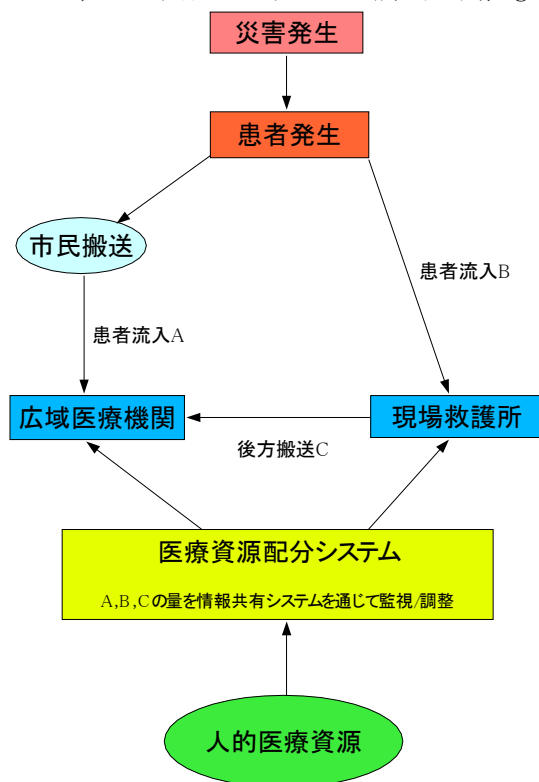


図 5.4-5 災害時における患者及び人的医療資源の流れ

- d : 治療時の平均容体急変時間
- e : 搬送時の平均容体急変時間
- f : 放置時の平均容体急変時間, $f < e < d$
- g : 放置時の平均死亡所要時間

t を行為開始後経過時間とし、患者にある行為をして t 時間後の状態遷移モデルは以下のようになる。

現場治療時の緊急治療患者の t 時間後の死亡確率は、

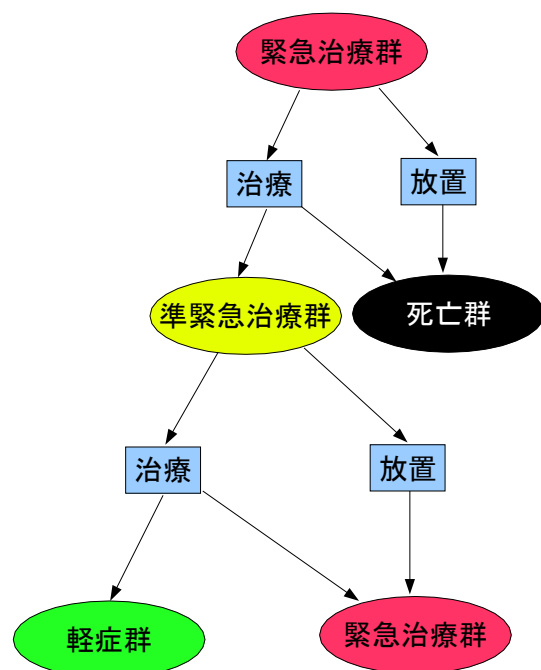


図 5.4-4 治療/放置によるトリアージ群間の確率的遷移モデル

$$P(B|R, FM, t) = \exp(-t/b)/b$$

で表され、時間 b 後も生存していれば、 Y に遷移する。

後方治療時の緊急治療患者の t 時間後の死亡確率は、

$$P(B|R, IM, t) = \exp(-t/c)/c$$

で表され、時間 c 後も生存していれば、 Y に遷移する。

後方治療時の準緊急治療患者の t 時間後の死亡確率は、

$$P(R|Y, IM, t) = \exp(-t/d)/d$$

で表され、時間 $2*d$ 後も容体安定していれば、 G に遷移する。

搬送時の準緊急治療患者の t 時間後に容体が緊急治療に遷移する確率は、

$$P(R|Y, T, t) = \exp(-t/e)/e$$

で表され、搬送終了後も容体安定していれば、 Y に遷移する。搬送所要時間はランダム変数で、

$$\text{Trans} = \text{rand}[1, \text{MAX}]$$

と表される。

放置時の緊急治療患者の t 時間後の死亡確率

$$P(B|R, L, t) = \exp(-t/g)/g$$

で表される。

放置時の準緊急治療患者の t 時間後に容体が緊急治療に遷移する確率は、

$$P(R|Y, L, t) = \exp(-t/f)/f$$

で表される。

以上のモデルは、トリアージによるおおまかな患者状態分類によって構築されているが、遷移モデルには一般性があり、典型的な外傷ごとに患者を分類して治療効果と予後の実データに基づき、より詳細なモデル化をすることも可能である。

d) 治療モデル

患者は、治療において状態遷移までの時間だけ医療資源を消費する。

e) シミュレーションについての検討

以上の外傷患者モデルを用い、現場救護所で治療をうける外傷患者数と、一般市民によって広域医療機関に搬送される外傷患者数の比率を変化させたとき、ある人的資源配分による救命率が影響を受けるかどうかを調べることが可能になった。自治体・医師会のデータを統合し、各医療機関の被害状況に基づく受け入れ容量に基づき、複数のシミュレーションを災害時に行うことで現場救護所と広域医療機関の最適な人的資源配分を調整することも可能である。救命率の評価が可能になることで、従来災害時に適切なトリアージについての見地が得られていなかった領域を検証することができる。

5.4.5 成果

(1) 口頭発表

- 野田五十樹：「統合シミュレーションのためのデータのバージョン管理の形式的考察」

FUN-AI 2005, 2005.3.7-9

(2) デモ展示

- マルチエージェントシミュレーションによる災害時の情報共有に関するデモ
IEEE HART & RoboCup Camp, 2004.12.4-6.

5.4.6 参考文献

- 1) 厚生省健康政策局計画課・厚生省健康政策局指導課監修, “災害時の地域保健医療活動”, 新企画出版社(1997/04)
- 2) 石原哲, “病院防災ガイドブックー災害発生時における病院防災対策のあり方”, 真興交易医書出版部(2001/02)
- 3) Robert L. Wears, Charles N. Winton, “SIMULATION MODELING OF PREHOSPITAL TRAUMA CARE”, Proceedings of the 1993 Winter Simulation Conference